REMARKS

The rejections presented in the Office Action dated June 7, 2004 have been considered.

Claim 1 is amended to include the limitations of claim 2, which is now canceled. Claim 9 is also amended.

Claim 10 is amended to correct a typographical error.

New claims 15-23 are added to claim other aspects of the invention.

Claims 1 and 3-23 remain pending in the application. Reconsideration and allowance of the application are respectfully requested.

The Office Action does not establish that claims 1 and 9 are unpatentable under 35 USC §103(a) over US patent 6,161,196 to Tsai (hereinafter, "Tsai") in view of "GDB Tutorial", by Kierstead et al., 1993 (hereinafter, "Kierstead"). The rejection is respectfully traversed because *prima facie* obviousness is not established. However, the rejection is now moot in view of the amended claims.

The Office Action does not establish that claims 2-8 and 10-14 are unpatentable under 35 USC §103(a) over Tsai and Kierstead, in further in view of US patent 5,590,277 to Fuchs (hereinafter, "Fuchs"). The rejection is respectfully traversed because the Office Action fails to show that all the limitations are suggested by the references, fails to provide a proper motivation for modifying the teachings of Tsai and Kierstead with teachings of Fuchs, and fails to show that the combination could be made with a reasonable likelihood of success.

The Office Action does not establish that the combination teaches or suggests the limitations of claim 2, which are now in claim 1. The limitations include, for example, initially re-executing the first set of object code upon detecting an error in execution of the first set of object code, and resuming execution using the second set of object code if the first set of object code fails in re-execution. The Office Action is mistaken in the allegation that Fuchs teaches these limitations. The cited section appears to teach a progressive recovery algorithm in which repeated faults of a process are counted and different recovery actions are taken based on the number of time the same fault occurs. Fuchs' final action appears to be to restart and reinitialize the system (col. 15, l. 48 – col. 16, l. 10; col. 17, ll. 26-31). There appears to be no apparent teaching of first attempting re-execution of the first set of object code, and upon failure resuming execution using the second set of object code from a second

compiler. No teaching by Fuchs of a second set of object code being from a second compiler is cited by the Office Action. Furthermore, no teaching of Tsai is cited to suggest these limitations. Therefore, the Office Action fails to show all the limitations of claim 2, now included in amended claim 1.

The alleged motivation for modifying Tsai with the teachings of Fuchs does not support *prima facie* obviousness. The alleged motivation simply states that this "modification would be obvious because one or ordinary skill in the art would be motivated to debug the faulty program, that is to recover the faulty program." No explanation is provided. Nor is it apparent how debugging a faulty program would be provided by including Fuch's progressive recovery algorithm in Tsai's multi-backend system. Furthermore, This alleged motivation is insufficient because it is merely a broad, conclusory statement of a speculative function. There is no evidence provided to indicate which elements of the Tsai-Kierstead combination could somehow be improved by specific elements of Fuchs. The alleged motivation lacks clear and particular reasons that would lead one of ordinary skill in the art to combine specific teachings of Fuchs with the Tsai-Kierstead combination.

Addressing the "rigorous ... requirement for a showing of the teaching or motivation to combine prior art references," the Court of Appeals for the Federal Circuit has stated:

> We have noted that evidence of a suggestion, teaching, or motivation to combine may flow from the prior art references themselves, the knowledge of one of ordinary skill in the art, or, in some cases, from the nature of the problem to be solved, (citations omitted), although "the suggestion more often comes from the teachings of the pertinent references," *Rouffet*, 149 F.3d at 1355, 47 USPQ2d at 1456. The range of sources available, however, does not diminish the requirement for actual evidence. That is, the showing must be clear and particular. *See, e.g., C.R. Bard*, 157 F.3d at 1352, 48 USPQ2d at 1232. Broad conclusory statements regarding the teaching of multiple references, standing alone, are not "evidence." (citation omitted) *In re Dembiczak*, 175 F.3d 994, 50 U.S.P.Q.2d 1614 (Fed. Cir. 1999).

The alleged motivation is merely a broad conclusory statement of general applicability, and no evidence is provided to suggest the combination. Therefore, the alleged motivation is insufficient to support *prima facie* obviousness.

The rejection further fails to show that the Tsai-Kierstead combination could be successfully modified with the teachings of Fuchs. The teachings of Tsai teach away from a successful modification with teachings of Fuchs. As Applicant previously argued, Tsai uses modular redundancy to provide fault tolerance (title) apparently because "other conventional schemes use algorithm-based detection methods that are generally not applicable to many

types of programs." (col. 1, ll. 61-63). Thus, the evidence suggests that Tsai teaches away from modification using an algorithm such as Fuchs' progressive recovery algorithm. The Office Action apparently responds to Applicant's argument by reasoning that since Tsai does not specifically mention Fuchs' approach by name, Tsai does not teach away from a modification from Fuchs. However, Applicant has provided specific evidence (by way of teachings of both Tsai and Fuchs) of why an algorithmic approach such as Fuchs' would not be desirable in Tsai, and the Office Action relies on conclusions instead of actual evidence. Thus, the evidence at hand suggests that Tsai teaches away from modification using an algorithm such as Fuchs' progressive recovery algorithm. For at least the reasons set forth above, *prima facie* obviousness is not established for claim 2 (now included with claim 1).

Claim 3 depends from claim 2 and includes further limitations that relate to re-executing the first set of object code a selected number of times before resuming execution using the second set of object code. As explained above, the Office Action fails to show that Fuchs resumes execution of any such second set of object code. Thus, even if Fuchs retries a faulty process, neither Fuchs nor the Tsai-Kierstead combination suggests resuming execution of a second set of object code which was generated from a second compiler.

Claim 4 includes further limitations that relate to the first and second sets of object code. Thus, for at least the reasons set forth above the Office Action does not establish a *prima facie* case of obviousness.

Claim 5 includes the limitations of claim 1 and further specifies selecting between the first and second sets of object code in resuming execution. The Office Action fails to show a teaching of these limitations by either Tsai or Fuchs. The apparent reasoning provided in the Office Action is, "Fuchs teaches selecting the first set of object code in resuming execution of the program (column 15, lines 48-67 to column 16, lines 1-10)", and the alleged motivation states, "it would have been obvious ... to modify the method disclosed by Tsai to include ... selecting the first set of object code in resuming execution of the program using the teaching of the combination of Kierstead and Fuchs [because] one of ordinary skill in the art would be motivated to debug the faulty program, that is to recover the faulty program." It is respectfully submitted that the Office Action appears to ignore the claimed aspect of selecting between the first and second sets of object code. No evidence is provided from either reference to suggest that both a first and a second set of object code are considered in making a selection. Thus, the Office Action does not show that either Tsai or Fuch "selects."

The Office Action has only shown that Tsai teaches executing multiple copies of a program, and Fuchs teaches increasing the scope of retried code based on the number of faults. No showing has been made that either of Tsai or Fuchs does any *selecting between* object code sets; Tsai executes all copies (no selecting needed), and Fuchs retries the same code (again, no selecting between two code sets). As explained above in regards to claim 2, the alleged motivation for combining Fuchs with the Tsai-Kierstead combination is improper. Therefore, *prima facie* obviousness is not established for claim 5.

Claims 6-8 depend directly or indirectly from claim 5 and include limitations comparable to those discussed above. Therefore, for at least the reasons set forth above the Office Action fails to establish a *prima facie* case obviousness for these claims.

Claim 10 is an apparatus claim, and claims 11-14 are to a computer program product. To the extent that claims 10 and 11-14 include limitations similar to those discussed above, *prima facie* obviousness is not established these claims.

The rejection of claims 2-8 and 10-14 over the Tsai-Kierstead-Fuchs combination should be withdrawn because the Office Action fails to show all the limitations are suggested by the combination, fails to provide a proper motivation for combining the references, and fails to show that the combination could be made with a reasonable likelihood of success.

Withdrawal of the rejection and reconsideration of the claims are respectfully requested in view of the remarks set forth above.

Respectfully submitted,

CRAWFORD MAUNU PLLC
1270 Northland Drive, Suite 390
Saint Paul, MN 55120
(651) 686-6633

By: 
Name: LeRoy D. Maunu
Reg. No.: 35,274